PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	G. Duncan Pearson	Confirmation No.	2080
Serial No.:	10/675,909		
Filed:	September 30, 2003	Customer No.:	51871
Examiner:	Tzu-Hsiang Lan		
Group Art Unit:	3623		
Docket No.:	1021-005US02		
Title:	NODE-LEVEL MODIFICATION DURING EXECUTION OF AN ENTERPRISE PLANNING MODEL		

CERTIFICATE UNDER 37 CFR 1.8 I hereby certify that this correspondence is being transmitted via the United States Patent and Trademark Office electronic filing system on January 4, 2011.

By:____/Patricia Cygan/______________________
Name: Patricia Cygan

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450,
Alexandria, VA 22313

Sir:

This Appeal Brief is responsive to the Notice of Panel Decision from Pre-Appeal Brief Review mailed on December 6, 2010. Appellant filed a Notice of Appeal accompanied by a Pre-Appeal Brief Request for Review on October 27, 2010 in response to the Final Office Action mailed on July 28, 1010. Because the Notice of Panel Decision was mailed December 6, 2010, the deadline for this Appeal Brief is January 6, 2011.

Please charge Deposit Account No. 50-1778 in the amount of $540.00 for Appellant's Appeal Brief fee for large entity.

Please charge any additional fees that may be required or credit any overpayment to Deposit Account No. 50-1778.

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corp. of Armonk, NY, the assignee of record.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

STATUS OF CLAIMS

Claims 7–10, 12, 13 and 22–31 are on appeal in this case. Claims 1–6, 11, 14–21 have been cancelled. Pending claims 7–10, 12, 13 and 22–31 are set forth in the attached Claims Appendix.

Claims 7–10, 12–13 and 22–31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Adaytum Software ("Adaytum") in view of Elkin et al. (U.S. Patent Publication No. 2007/0179828, hereinafter "Elkin"), in view of J. J. Halliday et al., "Flexible Workflow Management in the OPENflow System," (hereinafter, "Halliday") and further in view of Petra Heinl et al., "A Comprehensive Approach to Flexibility in Workflow Management Systems," (hereinafter, "Heinl").

STATUS OF AMENDMENTS

No amendments have been filed since the Final Office Action mailed on July 28, 1010.

SUMMARY OF CLAIMED SUBJECT MATTER

Claim 7 recites a method to perform node-level modification and maintenance of an enterprise planning model.[1] The method of claim 7 comprises executing, by a computing device,[2] an enterprise planning session[3] in accordance with an enterprise planning model,[4] wherein the enterprise planning model defines hierarchically arranged nodes[5] associated with business logic software modules[6] and enterprise contributors[7], wherein executing the enterprise planning session comprises: receiving, by the computing device contribution data[8] provided by the enterprise contributors, wherein the contribution data corresponds to one or more of the nodes of the enterprise planning model;[9] and automating, by the computing device, reconciliation[10] of the contribution data across an enterprise that corresponds to the enterprise planning model by automatically aggregating[11] the contribution data as the contribution data is received, wherein the enterprise planning model comprises a financial model;[12] checking-out,[13] by the computing device, an individual one of the nodes of the model for editing during execution of the enterprise planning session in accordance with the enterprise planning model; and modifying, by the computing device, the checked-out individual one of the nodes of the model without preventing execution of the enterprise planning session for the nodes of the enterprise planning model that are not checked out,[14] wherein at least one of the nodes of the

[1] *See, e.g.*, Appellant's specification, p. 11 ¶¶ 51,52; Figure 2, Abstract.

[2] *See, e.g.*, Appellant's specification, p. 7 ¶¶ 34, 37; p. 11 ¶ 52; p. 14 ¶ 62; Figure 3.

[3] *See, e.g.*, Appellant's specification, p. 6 ¶¶ 31, 32; p. 21 ¶ 93.

[4] *See, e.g.*, Appellant's specification, p. 6 ¶¶ 31, 32; p. 8 ¶¶ 39, 41; p. 10 ¶¶ 47, 48, 49; p. 16 ¶¶ 69, 70; p. 18 ¶ 75; Figures 5, 6, 20.

[5] *See, e.g.*, Appellant's specification, p. 5 ¶¶ 27, 28; p. 6 ¶ 32; p. 8 ¶ 39; p. 10 ¶ 47; p. 11 ¶ 51; p. 16 ¶ 69; p. 17 ¶¶ 71–74; p. 18 ¶ 77; p. 23 ¶ 107; Figures 5, 6.

[6] *See, e.g.*, Appellant's specification, p.8 ¶ 40; p. 9 ¶ 45; p. 11 ¶ 51; p. 16 ¶ 67; p. 21 ¶ 93; Figure 2.

[7] *See, e.g.*, Appellant's specification, p. 5 ¶¶ 28, 30; p. 6 ¶¶ 31, 32; p. 9 ¶¶ 42, 44; p. 11 ¶ 51; p. 13 ¶ 57; p. 15 ¶¶ 64, 65; p. 16 ¶ 69; p. 17 ¶ 71; p. 19 ¶ 82; p. 20 ¶ 86; Figures 1, 7, 8, 9.

[8] *See, e.g.*, Appellant's specification, p. 5 ¶ 30; p. 6 ¶ 31; p. 8 ¶ 41; p. 9 ¶¶ 42, 43; p. 10 ¶ 48; p. 11 ¶ 51; p. 13 ¶ 59; p. 15 ¶ 64; p. 16 ¶ 69; p. 19 ¶ 79 Figures 7, 8.

[9] *See, e.g.*, Appellant's specification, p. 5 ¶ 30; p. 6 ¶ 31; p. 8 ¶ 41; p. 9 ¶ 43; p. 10 ¶ 48; p. 11 ¶51; p. 16 ¶ 69; p. 17 ¶¶ 71, 73, 74; p. 19 ¶ 79.

[10] *See, e.g.*, Appellant's specification, p. 5 ¶¶ 27, 29; p. 6 ¶¶ 31, 33; p. 10 ¶ 48; p. 11 ¶ 52; p. 13 ¶ 59; p. 15 ¶ 65; p. 19 ¶ 80; Figure 1, 7.

[11] *See, e.g.*, Appellant's specification, p. 6 ¶¶ 31, 33; p. 9 ¶ 43; p. 15 ¶ 64; p. 17 ¶ 73; p. 19 ¶¶ 80, 82; p. 20 ¶ 83; Figures 7, 8.

[12] *See, e.g.*, Appellant's specification, p. 2 ¶ 7; p. 8 ¶¶ 41.

[13] *See, e.g.*, Appellant's specification, p. 11 ¶ 51.

[14] *See, e.g.*, Appellant's specification, p. 11 ¶ 51.

enterprise planning model that is not checked out receives contribution data from the checked-out individual one of the nodes without taking the model offline.[15]

Claim 24 recites a method comprising receiving, by a computing device,[16] an enterprise planning model[17] defining hierarchically arranged nodes[18] associated with business logic software modules[19] and enterprise contributors[20] of an enterprise, wherein the hierarchically arranged nodes comprise a first node and a second node,[21] and wherein the second node is hierarchically related to the first node;[22] associating, by the computing device, a first set of data with the first node and a second set of data with the second node in accordance with the enterprise planning model, such that the second set of data is hierarchically related to the first set of data;[23] receiving an update to the enterprise planning model, wherein the update identifies the first node;[24] checking-out, by the computing device, the first node after receiving the update to the enterprise planning model;[25] receiving, by the computing device, contribution data for the second node after checking-out the first node;[26] modifying, by the computing device, the second set of data with the received contribution data for the second node while the first node is checked-out;[27] checking-in, by the computing device, a modified version of the first node after modifying the second set of data for the second node, wherein the modified version of the first node corresponds to the received update to the enterprise planning model;[28] and reconciling, by the computing device, the modified second set of data with the first set of data of the modified version of the first node after the modified version of the first node has been checked in.[29]

[15] *See, e.g.*, Appellant's specification, p. 11 ¶ 51; p. 17–18 ¶ 74.

[16] *See, e.g.*, Appellant's specification, p. 7 ¶¶ 34, 37; p. 11 ¶ 52; p. 14 ¶ 62; Figure 3.

[17] *See, e.g.*, Appellant's specification, p. 6 ¶¶ 31, 32; p. 8 ¶¶ 39, 41; p. 10 ¶¶ 47– 49; p. 16 ¶¶ 69, 70; p. 18 ¶ 75; Figures 5, 6, 20.

[18] *See, e.g.*, Appellant's specification, p. 5 ¶¶ 27, 28; p. 6 ¶ 32; p. 8 ¶ 39; p. 10 ¶ 47; p. 11 ¶ 51; p. 16 ¶ 69; p. 17 ¶¶ 71–74; p. 18 ¶ 77; p. 23 ¶ 107; Figures 5, 6.

[19] *See, e.g.*, Appellant's specification, p.8 ¶ 40; p. 9 ¶ 45; p. 11 ¶ 51; p. 16 ¶ 67; p. 21 ¶ 93; Figure 2.

[20] *See, e.g.*, Appellant's specification, p. 5 ¶¶ 28, 30; p. 6 ¶¶ 31, 32; p. 9 ¶¶ 42, 44; p. 11 ¶ 51; p. 13 ¶ 57; p. 15 ¶¶ 64, 65; p. 16 ¶ 69; p. 17 ¶ 71; p. 19 ¶ 82; p. 20 ¶ 86; Figures 1, 7, 8, 9.

[21] *See, e.g.*, Appellant's specification, p. 5 ¶¶ 27, 28; p. 6 ¶ 32; p. 8 ¶ 39; p. 10 ¶ 47; p. 11 ¶ 51; p. 16 ¶ 69; p. 17 ¶¶ 71–74; p. 18 ¶ 77; p. 23 ¶ 107; Figures 5, 6.

[22] *Id.*

[23] *See, e.g.*, Appellant's specification, p.16 ¶ 69; p. 19 ¶ 82.

[24] *See, e.g.*, Appellant's specification, p. 11 ¶ 51, 52; p. 19 ¶ 79, 82.

[25] *See, e.g.*, Appellant's specification, p. 11 ¶ 51.

[26] *See, e.g.*, Appellant's specification, p. 11 ¶ 51.

[27] *See, e.g.*, Appellant's specification, p. 11 ¶ 51; p. 12 ¶ 52.

[28] *See, e.g.*, Appellant's specification, p. 11 ¶ 51; p. 12 ¶ 52.

[29] *See, e.g.*, Appellant's specification, p. 11 ¶ 51; p. 12 ¶ 52.

Claim 28 recites a method comprising a computer-readable storage medium encoded with instructions for causing a programmable processor to: receive an enterprise planning model[30] defining hierarchically arranged nodes[31] associated with business logic software modules[32] and enterprise contributors[33] of an enterprise, wherein the hierarchically arranged nodes comprise a first node and a second node,[34] and wherein the second node is hierarchically related to the first node;[35] associate a first set of data with the first node and a second set of data with the second node in accordance with the enterprise planning model, such that the second set of data is hierarchically related to the first set of data;[36] receive an update to the enterprise planning model, wherein the update identifies the first node;[37] check-out the first node after receiving the update to the enterprise planning model;[38] receive contribution data for the second node after checking-out the first node;[39] modify the second set of data with the received contribution data for the second node while the first node is checked-out;[40] check-in a modified version of the first node after modifying the second set of data for the second node, wherein the modified version of the first node corresponds to the received update to the enterprise planning model;[41] and reconcile the modified second set of data with the first set of data of the modified version of the first node after the modified version of the first node has been checked in.[42]

[30] *See, e.g.*, Appellant's specification, p. 6 ¶¶ 31, 32; p. 8 ¶¶ 39, 41; p. 10 ¶¶ 47– 49; p. 16 ¶¶ 69, 70; p. 18 ¶ 75; Figures 5, 6, 20.

[31] *See, e.g.*, Appellant's specification, p. 5 ¶¶ 27, 28; p. 6 ¶ 32; p. 8 ¶ 39; p. 10 ¶ 47; p. 11 ¶ 51; p. 16 ¶ 69; p. 17 ¶¶ 71–74; p. 18 ¶ 77; p. 23 ¶ 107; Figures 5, 6.

[32] *See, e.g.*, Appellant's specification, p.8 ¶ 40; p. 9 ¶ 45; p. 11 ¶ 51; p. 16 ¶ 67; p. 21 ¶ 93; Figure 2.

[33] *See, e.g.*, Appellant's specification, p. 5 ¶¶ 28, 30; p. 6 ¶¶ 31, 32; p. 9 ¶¶ 42, 44; p. 11 ¶ 51; p. 13 ¶ 57; p. 15 ¶¶ 64, 65; p. 16 ¶ 69; p. 17 ¶ 71; p. 19 ¶ 82; p. 20 ¶ 86; Figures 1, 7, 8, 9.

[34] *See, e.g.*, Appellant's specification, p. 5 ¶¶ 27, 28; p. 6 ¶ 32; p. 8 ¶ 39; p. 10 ¶ 47; p. 11 ¶ 51; p. 16 ¶ 69; p. 17 ¶¶ 71–74; p. 18 ¶ 77; p. 23 ¶ 107; Figures 5, 6.

[35] *Id.*

[36] *See, e.g.*, Appellant's specification, p.16 ¶ 69; p. 19 ¶ 82.

[37] *See, e.g.*, Appellant's specification, p. 11 ¶ 51, 52; p. 19 ¶ 79, 82.

[38] *See, e.g.*, Appellant's specification, p. 11 ¶ 51.

[39] *See, e.g.*, Appellant's specification, p. 11 ¶ 51.

[40] *See, e.g.*, Appellant's specification, p. 11 ¶ 51; p. 12 ¶ 52.

[41] *See, e.g.*, Appellant's specification, p. 11 ¶ 51; p. 12 ¶ 52.

[42] *See, e.g.*, Appellant's specification, p. 11 ¶ 51; p. 12 ¶ 52.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Appellant submits the following grounds of rejection to be reviewed on Appeal:

(1) Claims 7–10, 12–13 and 22–31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Adaytum Software ("Adaytum") in view of Elkin et al. (U.S. Patent Publication No. 2007/0179828, hereinafter "Elkin"), in view of J. J. Halliday et al., "Flexible Workflow Management in the OPENflow System," (hereinafter, "Halliday") and further in view of Petra Heinl et al., "A Comprehensive Approach to Flexibility in Workflow Management Systems," (hereinafter, "Heinl").

ARGUMENT

There is one ground of rejection under appeal. The ground of rejection under appeal is the rejection of claims 7–10, 12–13 and 22–31 under 35 U.S.C. § 103(a) as being unpatentable over Adaytum Software ("Adaytum") in view of Elkin et al. (U.S. Patent Publication No. 2007/0179828, hereinafter "Elkin"), in view of J. J. Halliday et al., "Flexible Workflow Management in the OPENflow System," (hereinafter, "Halliday") and further in view of Petra Heinl et al., "A Comprehensive Approach to Flexibility in Workflow Management Systems," (hereinafter, "Heinl"). Appellant respectfully traverses the current rejections advanced in the Final Office Action, and requests reversal by the Board of Patent Appeals based on the arguments below. Appellant respectfully requests review by the Board for Group 1 addressed below.

The Patent Examiner bears the burden of proof to demonstrate a prima facie case that an invention is not patentable.[43] In reviewing an Examiner's decision on Appeal, the Board must consider all of the evidence, and patentability is determined by a preponderance of the evidence with due consideration to persuasiveness of argument.[44]

The Supreme Court recently clarified the standard of non-obviousness under 35 U.S.C. 103(a) in *KSR Int'l Co. v. Teleflex, Inc.*[45] As reiterated by the Supreme Court in *KSR International Co.* v. *Teleflex Inc.* (KSR),[46] the framework for the objective analysis for determining obviousness under 35 U.S.C. 103 is stated in *Graham* v. *John Deere Co.*[47]

[43] *See In re Oetiker,* 977 F.2d 1443.
[44] *Id.*
[45] *See KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 82 USPQ2d 1385 (2007).
[46] 82 USPQ2d 1385.
[47] 383 U.S. 1, 148 USPQ 459.

Obviousness is a question of law based on underlying factual inquiries. The factual inquiries enunciated by the Court are as follows:

(1) Determining the scope and content of the prior art;

(2) Ascertaining the differences between the claimed invention and the prior art; and

(3) Resolving the level of ordinary skill in the pertinent art.

In *KSR*, the Supreme Court explained that the Examiner must provide an articulated reasoning with some rational underpinning why a person of ordinary skill in the art would have been led to make a modification or combination to arrive at the claimed invention. An invention composed of several elements is not proved obvious merely by demonstrating that each of the elements was independently known.[48]

Consistent with *KSR*, the Federal Circuit has stated that there must be "some rationale, articulation, or reasoned basis" to support the legal conclusion of obviousness."[49] The reason for modification need not conform to the particular motivation or objective of the patent Appellant.[50] However, there still must be some need or problem known in the art that would have provided a reason for combining elements in the manner claimed.[51]

Furthermore, a basic premise of the obviousness analysis set forth in *KSR* is that the combination of prior art references must actually disclose the elements recited in the claims. Consistent with this premise, the Manual for Patenting Examining Procedure (MPEP) sets forth three examples of rationales that may support a conclusion of obviousness.[52] As one example, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. As another example, there must be a reasonable expectation of success. As yet another example, the prior art reference (or references when combined) must teach or suggest all the claim limitations.[53]

[48] 82 USPQ2d 1396.

[49] *Alza Corp. v. Mylan Laboratories*, 80 USPQ2d 1001, 1005 (Fed. Cir. 2006) (citing *In re Kahn*, 78 USPQ2d 1329 (Fed. Cir. 2006)).

[50] 82 USPQ2d 1397.

[51] *Id.*

[52] See MPEP 2143.

[53] See MPEP 2143.

The *KSR* case clarified that the "suggestion or motivation" requirement is more broadly a requirement that the Examiner articulate an "apparent reason" for the modification. However, the *KSR* case did not modify the basic requirement of the obviousness analysis that requires the Examiner to show that the prior art collectively teaches the elements of Appellant's claims. Accordingly, if Appellant can show that the prior art lacks a teaching of one or more elements of the pending claims, the obviousness rejections must be reversed. In addition, if the Examiner has not provided some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness, the obviousness rejections must be reversed.

FIRST GROUND OF REJECTION UNDER APPEAL

GROUP 1—(7–10, 12–13 and 22–31)

In the Final Office Action, the Examiner rejected claims 7–10, 12–13 and 22–31 under 35 U.S.C. § 103(a) as being unpatentable over Adaytum in view of Elkin, Halliday, and Heinl. Appellant respectfully traverses the rejection. The applied references fail to disclose or suggest the features defined by Appellant's claims, and provide no apparent reason for modification to arrive at the claimed features. Appellant notes as a preliminary matter that the Final Office Action cited Du et al. (U.S. Patent No. 6,308,163 hereinafter "Du") as allegedly showing that hierarchical enterprise structure for enterprise planning of claim 7 is old and well known.[54] Appellant does not acquiesce to this assertion of the Final Office Action. However, even assuming *arguendo* that Du did disclose a hierarchical enterprise structure for enterprise planning, the combination of Du with Adaytum, Elkin, Halliday, and Heinl, would still not disclose or suggest the requirements of the pending claims. Appellant's claims recite elements not disclosed by any of Adaytum, Elkin, Halliday, Heinl, and Du, alone or in any combination.

Appellant's claim 7, for example, requires modifying, by a computing device, a checked-out individual one of nodes of an enterprise planning model without preventing execution of an enterprise planning session for the nodes of the enterprise planning model that are not checked-out, <u>wherein at least one of the nodes of the enterprise planning model that are not checked-out receives contribution data from the checked-out individual one of the nodes without taking the model offline</u>. *See, e.g.*, Appellant's specification, ¶¶ [051], [074]. Elkin in view of Adaytum Software, Heinl, and Halliday fails to disclose or suggest the requirements of claim 7.

[54] *See* Final Office Action, dated July 28, 2010, p. 5.

The Final Office Action asserted that "the node level operation for online node to receive data from a checked-out node is readily disclosed by Heinl." Final Office Action dated July 28, 2010, p. 4 ¶1. In support of this assertion, the Office Action cited Heinl, p. 85 col. 2 ¶1 and p. 86 col. 2 ¶5, "where allowing 'dirty reads' one or more of online nodes will be able to read and receive partial data from a checked-out node." Appellant respectfully submits, however, that Heinl fails to disclose or suggest wherein at least one of the nodes of the enterprise planning model is are not checked out receives contribution data from the checked-out individual one of the nodes without taking the model offline, as required by claim 7.

Heinl generally discloses checking out a workflow type such that only one person can edit the workflow type, and thus that this workflow type is locked with respect to a second person. Heinl, § 4.2.2. Appellant respectfully submits that a workflow type, as described by Heinl, does not disclose or suggest a node of an enterprise planning model. An enterprise planning model, as required by claim 7, "defines hierarchically arranged nodes associated with business logic software modules and enterprise contributors." Workflow types of Heinl, on the other hand, "model business processes . . . . A workflow type may be instantiated in order to represent a performing occurrence of a business process." Heinl, § 1 (emphasis added). Accordingly, to the extent that Heinl may disclose modifying a checked out workflow type, such disclosure is not relevant to the requirements of Appellant's claim 7, which requires modifying a checked-out individual one of nodes of an enterprise planning model.

The Final Office action further confuses the distinction between a workflow model and an enterprise planning model by incorrectly reading the language of claim 7 to require a "workflow model." Final Office Action dated July 28, 2010, p. 10. Claim 7 requires an "enterprise planning model" not a "workflow model" and therefore the disclosure of Heinl is not relevant to Appellant's claim 7.

To the extent that Adaytum may disclose nodes of an enterprise planning model, the applied references do not disclose or suggest how to apply the techniques of checking out a workflow type, as disclosed by Heinl, to a node of an enterprise planning model. Because a workflow type specifies a suitable execution path to travel from a start point to an end point,[55] a workflow type of Heinl is not associated with a set of data and cannot receive contribution data,

[55] Heinl, p. 81, col. 1, § 2.1 (stating that "suitable execution paths" for coming "from the start to the end point" are "directly specified in the workflow type").

contrary to the requirements of claim 7. Thus, the techniques described by Heinl with respect to checking out a workflow type cannot be readily combined with the disclosure of Adaytum to arrive at the requirements of claim 7.

Even if Heinl did disclose modifying a checked-out node of a model, to which Appellant does not acquiesce, Heinl still fails to disclose or suggest that at least one node of an enterprise planning model that is not checked-out receives contribution data from the checked-out individual one of the nodes without taking the model offline, as required by claim 7. In Heinl, there is no interdependency between workflow types. That is, one workflow type cannot receive data from a workflow type that is checked out while the checked-out workflow type is checked out. At most, a modeler is able to receive a specification for a workflow type to edit a separate workflow type specification. Heinl, § 4.2.2,p. 85, col. 2, ¶ 1. Editing a separate workflow type specification based on a checked-out workflow type specification does not disclose or suggest at least one node of an enterprise planning model that is not checked out that receives contribution data from a checked out individual one of the nodes of the model, without taking the model offline, as required by claim 7.

To be clear, workflow types of Heinl specify execution paths to travel from one point to another. Workflow types of Heinl, as noted above, do not store contribution data. In other words, to the extent that Heinl may disclose referencing a work type specification by concurrent modelers (*e.g.*, Heinl, p. 85, col. 2), Heinl does not disclose or suggest receiving, with a non-checked-out node, contribution data from a checked out node of an enterprise planning model while the checked-out node is checked out. An enterprise planning model defines hierarchically arranged nodes, as required by claim 7. In addition, nodes of the enterprise planning model are associated with contribution data. Thus, in the context of claim 7, there are two types of data: 1) the definition of the model, and 2) contribution data associated with a node of the model.

The Final Office Action fails give proper weight to the distinction between the definition of the model and the contribution data. Claim 7 distinctly claims and separately requires a node and contribution data. The Final Office Action's argument relies on construing a workflow type as both a node and as contribution data. For example, the Final Office Action improperly argues that "'contribution data' is merely a non-functional descriptive label" and "as long as one node of [the] enterprise model that is not checked out [is] able to receive **any** data from a checked out node without taking model offline it meets the claim." Final Office Action dated July 28, 2010, p. 4.

Appellant respectfully notes that during patent examination, claims must be given their "broadest reasonable interpretation consistent with the specification".[56] Moreover, when two different words are used in the same claim, they are required to have different meanings.[57] The Final Office Action attempts to read away the distinct meanings of "contribution data" and "node" by construing contribution data as "**any** data" and therefore equating "contribution data" and "node." Having set aside any distinction between "contribution data" and "node," the Final Office Action equates a first workflow type receiving a second workflow type to a node receiving contribution data. While a first and a second workflow type are each a workflow type, a node and contribution data cannot be similarly equated.[58] For example, equating a node to contribution data would lead to nonsensical interpretations of claim 7, such as contribution data receiving contribution data. Such interpretations are not consistent with interpretations that those skilled in the art would reach.[59] Moreover, workflow types of Heinl do not have associated contribution data, contrary to the requirements of claim 7. Thus, Heinl (even in view of the other applied references) fails to disclose or suggest a node of an enterprise planning model that is not checked out that receives contribution data from a checked-out individual one of the nodes as required by claim 7.

Halliday also fails to disclose or suggest the requirements of claim 7, e.g., that a node of an enterprise planning model that is not checked out that receives contribution data from a checked-out individual one of the nodes without taking the model offline. Similar to Heinl, Halliday is also directed to a workflow model. Halliday, Abstract. The Final Office Action cited page 7 of Halliday, asserting that tasks are individual nodes of workflow. Final Office Action dated July 28, 2010, p. 11. However, tasks of a workflow model, as discussed above with respect to Heinl, are not the same as nodes of an enterprise planning model. For example, tasks are not associated with contribution data, as required by Appellant's claim 7. Thus, to the extent that Halliday may disclose changing tasks of a workflow model, such disclosure would not disclose or suggest the requirements of Appellant's claim 7, even in view of the other applied references.

[56] *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005) (en banc) (emphasis added); MPEP § 2111.

[57] *Primos, Inc. v. Hunter's Specialties, Inc.*, 451 F.3d 841, 847–848 (Fed. Cir. 2006).

[58] *Id.*

[59] *In re Cortright*, 165 F.3d 1353, 1359 (Fed. Cir. 1999); MPEP § 2111.

In addition, Halliday states that changes to a workflow model "must be performed consistently . . . to a workflow schema instance [and] are carried out atomically (either all changes are performed or none) with respect to the normal processing activities." Halliday, p. 7, § 2.3. A system according to Halliday cannot provide a node of an enterprise planning model that is not checked out that receives contribution data from a checked-out individual one of the nodes without taking the model offline, because all updates in the system according to Hailliday are made at the same time. For at least these reasons, independent claim 7 is patentable over the applied references. Independent claims 24 and 28 include similar requirements, for which similar remarks apply. Therefore, claims 24 and 28 are also patentable over the applied references. The dependent claims, i.e., claims 8–10, 12, 13, 22–23, 25–27 and 29–31, incorporate the subject matter of the respective independent claims.[60] Accordingly, for at least the reasons discussed above, all of the pending claims are patentable over the applied references under 35 U.S.C. § 103(a). Appellant respectfully requests reversal of this rejection.

CONCLUSION OF ARGUMENT

In view of Appellant's arguments, the final rejections of Appellant's claims are improper and should be reversed. Reversal of all pending rejections and allowance of all pending claims is respectfully requested.

Date: January 4, 2011

By: /Christopher D. Karlen, Reg. # 66,486/
Name: Christopher D. Karlen

SHUMAKER & SIEFFERT, P.A.
1625 Radio Drive, Suite 300
Woodbury, Minnesota 55125
Telephone: 651.286.8390
Facsimile: 651.735.1102

[60] 35 U.S.C. § 112, ¶ 4.

APPENDIX: CLAIMS ON APPEAL

Claims 1–6 (Cancelled).

Claim 7 (Previously Presented): A computer-implemented method comprising:

executing, by a computing device, an enterprise planning session in accordance with an enterprise planning model, wherein the enterprise planning model defines hierarchically arranged nodes associated with business logic software modules and enterprise contributors, wherein executing the enterprise planning session comprises:

receiving, by the computing device contribution data provided by the enterprise contributors, wherein the contribution data corresponds to one or more of the nodes of the enterprise planning model; and

automating, by the computing device, reconciliation of the contribution data across an enterprise that corresponds to the enterprise planning model by automatically aggregating the contribution data as the contribution data is received, wherein the enterprise planning model comprises a financial model;

checking-out, by the computing device, an individual one of the nodes of the model for editing during execution of the enterprise planning session in accordance with the enterprise planning model; and

modifying, by the computing device, the checked-out individual one of the nodes of the model without preventing execution of the enterprise planning session for the nodes of the enterprise planning model that are not checked out, wherein at least one of the nodes of the enterprise planning model that is not checked out receives contribution data from the checked-out individual one of the nodes without taking the model offline.

Claim 8 (Previously Presented): The computer-implemented method of claim 7, wherein modifying the checked-out individual one of the nodes comprises:

receiving updated model information for the checked-out individual one of the nodes; and

updating a respective slice of the enterprise planning model for only the checked-out individual one of the nodes based on the updated model information.

Claim 9 (Previously Presented): The computer-implemented method of claim 8, wherein updating the enterprise planning model comprises modifying the business logic software module or the enterprise contributor associated with the checked-out individual one of the nodes in response to the updated model information.

Claim 10 (Previously Presented): The computer-implemented method of claim 8:

wherein executing the enterprise planning session comprises receiving and processing the contribution data from the enterprise contributors associated with the nodes of the model during the execution of the enterprise planning session and prior to the check-out of the individual one of the nodes; and

wherein modifying the checked-out individual one of the nodes comprises updating data of the checked-out one of the nodes with the contribution data that was received prior to the check-out of the individual one of the nodes in accordance with the updated model information when the checked-out individual one of the nodes is subsequently checked-in during the execution of the enterprise planning session.

Claim 11 (Cancelled).

Claim 12 (Previously Presented): The computer-implemented method of claim 10, wherein automating reconciliation comprises defining reconciliation jobs for execution by an application server to prompt a reviewer to reconcile the previously received contribution data with the updated model information for the checked-in individual one of the nodes, wherein the application server is communicatively coupled to the computing device.

Claim 13 (Previously Presented): The computer-implemented method of claim 10, wherein automating reconciliation comprises defining reconciliation jobs for execution by remote computers of the enterprise contributors to prompt at least one of the enterprise contributors to reconcile the previously received contribution data with the updated model information for the checked-in individual one of the nodes.

Claims 14–21 (Cancelled).

Claim 22 (Previously Presented): The method of claim 7, wherein automatically aggregating the contribution data as the contribution data is received comprises:

receiving a portion of the contribution data;

identifying higher levels of the hierarchically arranged nodes affected by the portion of the contribution data; and

calculating new aggregate totals at each level of the hierarchically arranged nodes in real time according to the received portion.

Claim 23 (Previously Presented): The method of claim 12, further comprising receiving an indication from the reviewer corresponding to the checked-in individual one of the nodes, wherein the indication indicates whether the reviewer accepted or rejected the contribution data for the checked-in individual one of the nodes.

Claim 24 (Previously Presented): A method comprising:

receiving, by a computing device, an enterprise planning model defining hierarchically arranged nodes associated with business logic software modules and enterprise contributors of an enterprise, wherein the hierarchically arranged nodes comprise a first node and a second node, and wherein the second node is hierarchically related to the first node;

associating, by the computing device, a first set of data with the first node and a second set of data with the second node in accordance with the enterprise planning model, such that the second set of data is hierarchically related to the first set of data;

receiving an update to the enterprise planning model, wherein the update identifies the first node;

checking-out, by the computing device, the first node after receiving the update to the enterprise planning model;

receiving, by the computing device, contribution data for the second node after checking-out the first node;

modifying, by the computing device, the second set of data with the received contribution data for the second node while the first node is checked-out;

checking-in, by the computing device, a modified version of the first node after modifying the second set of data for the second node, wherein the modified version of the first node corresponds to the received update to the enterprise planning model; and

reconciling, by the computing device, the modified second set of data with the first set of data of the modified version of the first node after the modified version of the first node has been checked in.

Claim 25 (Previously Presented): The method of claim 24, wherein the first node comprises a first child node, wherein the second node comprises a second child node, wherein the hierarchically arranged nodes further comprise a parent node, wherein the enterprise planning model defines the parent node as a parent to the first child node and the second child node, the method further comprising:

aggregating, after checking-in the modified version of the first child node, the first data set for the modified version of the first child node and the second data set comprising the received contribution data for the second child node to form a set of aggregate data; and

associating the set of aggregate data with the parent node.

Claim 26 (Previously Presented): The method of claim 24, further comprising:

receiving a second set of contribution data for the first node before checking-out the first node;

defining a reconciliation job for execution by an application server that is communicatively coupled to the computing device, wherein the reconciliation job is configured to cause the application server to prompt a reviewer to reconcile the second set of contribution data with the modified version of the first node; and

receiving, by the computing device, a response from the application server indicating whether the reviewer has accepted or rejected the second set of contribution data for the modified version of the first node.

Claim 27 (Previously Presented): The method of claim 24, further comprising updating a slice of the enterprise planning model corresponding to the first node to form the modified version of the first child node while the first child node is checked out.

Claim 28 (Previously Presented): A computer-readable storage medium encoded with instructions for causing a programmable processor to:

receive an enterprise planning model defining hierarchically arranged nodes associated with business logic software modules and enterprise contributors of an enterprise, wherein the hierarchically arranged nodes comprise a first node and a second node, and wherein the second node is hierarchically related to the first node;

associate a first set of data with the first node and a second set of data with the second node in accordance with the enterprise planning model, such that the second set of data is hierarchically related to the first set of data;

receive an update to the enterprise planning model, wherein the update identifies the first node;

check-out the first node after receiving the update to the enterprise planning model;

receive contribution data for the second node after checking-out the first node;

modify the second set of data with the received contribution data for the second node while the first node is checked-out;

check-in a modified version of the first node after modifying the second set of data for the second node, wherein the modified version of the first node corresponds to the received update to the enterprise planning model; and

reconcile the modified second set of data with the first set of data of the modified version of the first node after the modified version of the first node has been checked in.

Claim 29 (Previously Presented): The computer-readable storage medium of claim 28, wherein the first node comprises a first child node, wherein the second node comprises a second child node, wherein the hierarchically arranged nodes further comprise a parent node, wherein the enterprise planning model defines the parent node as a parent to the first child node and the second child node, further comprising instructions to:

aggregate, after checking-in the modified version of the first child node, the first data set for the modified version of the first child node and the second data set comprising the received contribution data for the second child node to form a set of aggregate data; and

associate the set of aggregate data with the parent node.

Claim 30 (Previously Presented): The computer-readable storage medium of claim 28, further comprising instructions to:

receive a second set of contribution data for the first node before checking-out the first node;

define a reconciliation job for execution by an application server that is communicatively coupled to the computing device, wherein the reconciliation job is configured to cause the application server to prompt a reviewer to reconcile the second set of contribution data with the modified version of the first node; and

receive, by the computing device, a response from the application server indicating whether the reviewer has accepted or rejected the second set of contribution data for the modified version of the first node.

Claim 31 (Previously Presented): The computer-readable storage medium of claim 28, further comprising instructions to update a slice of the enterprise planning model corresponding to the first node to form the modified version of the first child node while the first child node is checked out.

APPENDIX: EVIDENCE

None

APPENDIX: RELATED PROCEEDINGS

None